

## Trabalho 02 - Problema da k-Seleção

Data de entrega: 31/05/2024

Importante:

- **Não** olhe códigos de outros grupos ou da internet. Exceto o que é fornecido.
- O trabalho pode ser feito em duplas.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes do `runcodes.hokama.com.br` não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Você deverá submeter, até a data de entrega, o seu código na plataforma `runcodes.hokama.com.br` (se estiver em dupla, apenas 1 deve submeter).
- Coloque o(s) número(s) de matrícula na primeira linha do código.

Neste trabalho você deverá implementar um Algoritmo para o Problema da k-Seleção, onde é dado um arranjo  $A$  com  $n$  números inteiros e um arranjo ordenado  $i$  com  $k$  números entre 1 e  $n$ . O objetivo é encontrar o  $i[0]$ -ésimo menor elemento de  $A$ , o  $i[1]$ -ésimo menor elemento de  $A$ , ..., até o  $i[k-1]$ -ésimo menor elemento de  $A$ .

Por exemplo: Se  $A = \{10, 2, 3, 100, 6\}$  e  $i = \{1, 2, 5\}$ , a solução para essa instância é composta pelo primeiro menor valor, o segundo menor valor e quinto menor valor, portanto, os valores 2, 3 e 100. (Note que aqui, **diferente** do problema visto em sala, **1 significa o menor**)

Neste trabalho o seu programa deverá ler dois inteiros, que representam a quantidade  $n$  de valores que serão gerados aleatoriamente para formar  $A$ , e a quantidade  $k$  de estatísticas de ordem que serão geradas para formar  $i$ . Um código que já faz isso será fornecido, sugiro não mudar essa parte do programa pois é preciso garantir que os números gerados sejam sempre os mesmos.

Veja um exemplo de entrada:

|      |
|------|
| 11 4 |
|------|

O seu programa então vai gerar 11 valores aleatórios e 4 estatísticas de ordem. Nesse caso o arranjo  $A$  gerado será (250080677635300, 142390661764278, 217695086224632, 44211319422675, 163690663340080, 172797822383978, 33812607614009, 72043229582031, 132196207797160, 16989489435043, 18811839658455) e o vetor  $i$  gerado será [3, 4, 10, 10] portanto seu programa deverá imprimir, o terceiro menor elemento, depois o quarto e o décimo menor elemento (duas vezes):

33812607614009 44211319422675 217695086224632 217695086224632

Você receberá um arquivo *kSelect.py* que contém uma implementação que ordena o vetor *A* com o Insertion Sort e imprime  $A[i][j]-1$  para cada *j* entre 0 e *k*-1. Infelizmente ele não consegue resolver os testes em menos de 1 segundo, o que o seu algoritmo deve fazer. Você deve manter a forma como ele sorteia os valores.

**Você não pode** usar a função **sort** do python, ou similar, já que essa é implementada em **C**.

- Você deverá implementar em linguagem python.
- Seu programa deve executar no [runcodes.hokama.com.br](http://runcodes.hokama.com.br) em menos de 1 segundo.
- Você não deve usar nenhuma função pronta muito complexa.
- Se você não tiver certeza se alguma coisa é permitida ou não no trabalho, não hesite em perguntar ao professor!
- Não deixe para os últimos dias!